



▶ **Displaying CFD Solution Parameters on Arbitrary Cut Planes**

Langley Research Center, Hampton, Virginia

USMC6 is a Fortran 90 computer program for post-processing in support of visualization of flows simulated by computational fluid dynamics (CFD). The name "USMC6" is partly an abbreviation of "TetrUSS — USM3D Solution Cutter," reflecting its origin as a post-processor for use with USM3D — a CFD program that is a component of the Tetrahedral Unstructured Software System and that solves the Navier-Stokes equations on tetrahedral un-

structured grids. "Cutter" here refers to a capability to acquire and process solution data on (1) arbitrary planes that cut through grid volumes, or (2) user-selected spheroidal, conical, cylindrical, and/or prismatic domains cut from within grids. Cutting saves time by enabling concentration of post-processing and visualization efforts on smaller solution domains of interest.

The user can select from among more than 40 flow functions. The cut planes

can be trimmed to circular or rectangular shape. The user specifies cuts and functions in a free-format input file using simple and easy-to-remember keywords. The USMC6 command line is simple enough that the slicing process can readily be embedded in a shell script for assembly-line post-processing. The output of USMC6 is a data file ready for plotting.

This program was written by S. Paul Pao of Langley Research Center. LAR-17527-1

▶ **Flow Solver for Incompressible 2-D Drive Cavity**

Goddard Space Flight Center, Greenbelt, Maryland

This software solves the Navier-Stokes equations for the incompressible driven cavity flow problem. The code uses second-order finite differencing on a staggered grid using the Chorin projection method. The resulting intermediate Poisson equation is

efficiently solved using the fast Fourier transform.

Time stepping is done using fourth-order Runge-Kutta for stability at high Reynolds numbers. Features include check-pointing, periodic field snapshots, ongoing reporting of kinetic energy and changes between time

steps, time histories at selected points, and optional streakline generation.

This program was written by Virginia Kalb of Goddard Space Flight Center. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-15107-1

▶ **Flow Solver for Incompressible Rectangular Domains**

Goddard Space Flight Center, Greenbelt, Maryland

This is an extension of the Flow Solver for Incompressible 2-D Drive Cavity software described in the preceding article. It solves the Navier-Stokes equations for incompressible flow using finite differencing on a uniform, staggered grid. There is a runtime choice of either central differencing or modified upwinding for the convective term. The domain must be rectangular, but may have a rectangular walled region within it. Currently, the position of the interior region and exterior

boundary conditions are changed by modifying parameters in the code and recompiling. These features make it possible to solve a variety of classical fluid flow problems such as an L-shaped cavity, channel flow, or wake flow past a square cylinder. The code uses fourth-order Runge-Kutta time-stepping and overall second-order spatial accuracy.

This software permits the walled region to be positioned such that flow past a square cylinder, an L-shaped cavity, and

the flow over a back-facing step can all be solved by reconfiguration. Also, this extension has an automatic detection of periodicity, as well as use of specialized data structure for ease of configuring domain decomposition and computing convergence in overlap regions.

This program was written by Virginia L. Kalb of Goddard Space Flight Center. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-15111-1

▶ **Simulating Avionics Upgrades to the Space Shuttles**

Lyndon B. Johnson Space Center, Houston, Texas

Cockpit Avionics Prototyping Environment (CAPE) is a computer program that simulates the functions of proposed

upgraded avionics for a space shuttle. In CAPE, pre-existing space-shuttle-simulation programs are merged with a

commercial-off-the-shelf (COTS) display-development program, yielding a package of software that enables high-fi-

delity simulation while making it possible to rapidly change avionic displays and the underlying model algorithms. The pre-existing simulation programs are Shuttle Engineering Simulation, Shuttle Engineering Simulation II, Interactive Control and Docking Simulation, and Shuttle Mission Simulator playback.

The COTS program — Virtual Application Prototyping System (VAPS) —

not only enables the development of displays but also makes it possible to move data about, capture and process events, and connect to a simulation. VAPS also enables the user to write code in the C or C++ programming language and compile that code into the end-product simulation software. As many as ten different avionic-upgrade ideas can be incorporated in a single compilation

and, thus, tested in a single simulation run. CAPE can be run in conjunction with any or all of four simulations, each representing a different phase of a space-shuttle flight.

This program was written by Daniel Deger and Kenneth Hill of Johnson Space Center and Karsten E. Braaten of United Space Alliance. Further information is contained in a TSP (see page 1). MSC-23453-1/15-1

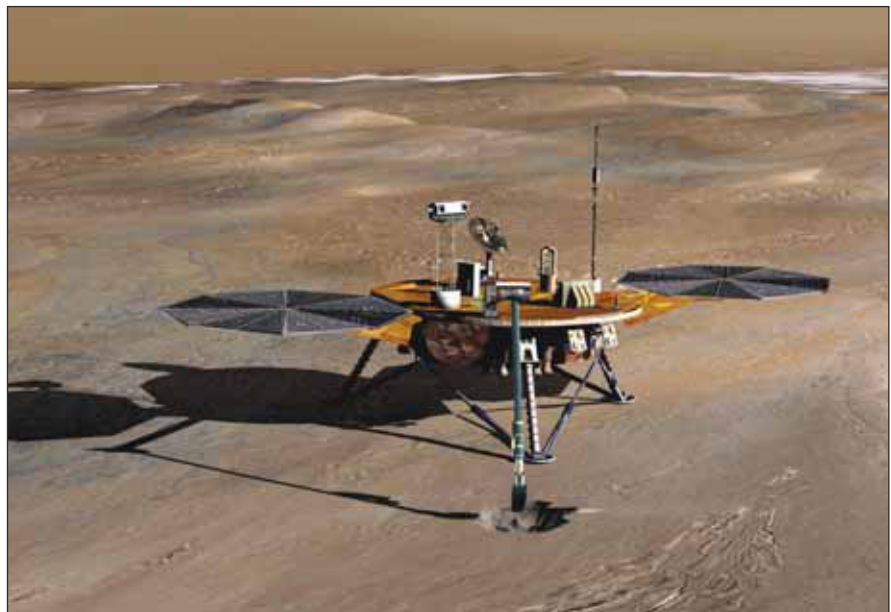
Simulating the Phoenix Landing Radar System

NASA's Jet Propulsion Laboratory, Pasadena, California

A computer program called “phxlr-sim” simulates the behavior of the radar system used as an altimeter and velocimeter during the entry, descent, and landing phases of the Phoenix lander spacecraft. The simulation includes modeling of internal functions of the radar system, the spacecraft trajectory, and the terrain. The computational models incorporate representations of nonideal hardware effects in the radar system and effects of radar speckle (coherent scatter of radar signals from terrain).

This program was written by Curtis W. Chen of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44431.



Phoenix Mission Lander on Mars, artist's concept.

Injecting Artificial Memory Errors Into a Running Computer Program

NASA's Jet Propulsion Laboratory, Pasadena, California

Single-event upsets (SEUs) or “bit-flips” are computer memory errors caused by radiation. BITFLIPS (Basic Instrumentation Tool for Fault Localized Injection of Probabilistic SEUs) is a computer program that deliberately injects SEUs into another computer program, while the latter is running, for the purpose of evaluating the fault tolerance of that program. BITFLIPS was written as a plug-in extension of the open-source Valgrind debugging and profiling software. BITFLIPS can inject SEUs into any program that can

be run on the Linux operating system, without needing to modify the program's source code. Further, if access to the original program source code is available, BITFLIPS offers fine-grained control over exactly when and which areas of memory (as specified via program variables) will be subjected to SEUs.

The rate of injection of SEUs is controlled by specifying either a fault probability or a fault rate based on memory size and radiation exposure time, in units of SEUs per byte per

second. BITFLIPS can also log each SEU that it injects and, if program source code is available, report the magnitude of effect of the SEU on a floating-point value or other program variable.

This program was written Benjamin J. Bornstein, Robert A. Granat, and Kiri L. Wagstaff of Caltech for NASA's Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-45368.